文本自动生成及解码算法介绍

矣晓沅

THUNLP Lab

清华大学 人工智能研究院

OUTLINE

OUTLINE

01 文本生成任务简介

Tasks

1. Natural Language Understanding (NLU)

   e.g., sentiment analysis, text similarity, paraphrase detection (GLUE Benchmark)

2. Natural Language Generation (NLG)

- Machine Translation
- Response Generation
- Generative QA
- Generative Summarization
- Review Generation
- Image / Video Captioning
- Narrative Generation / Story Telling
- Poetry Generation

……

01 文本生成任务简介

Overview & Formulation

1. Autoregressive Generation

- Generic Text Generation

x a sequence with n tokens $x = (x_1, \ldots, x_n)$

Modelling: $p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i | x_{<i})$

Dataset: $D = \{x^1, \ldots, x^{|D|}\}$ → Loss: $L(\theta) = -\sum_{i=k}^{|D|} \log p_\theta(x_i^k | x_{<i}^k)$

Decoding: $x^* = \underset{x}{\text{argmax}} \sum_i \log p_\theta(x_i | x_{<i})$

01 文本生成任务简介

Overview & Formulation

1. Autoregressive Generation

- Conditional Text Generation

$x = (x_1, \dots, x_n)$ condition c Modelling: $p_\theta(x|c) = \prod_{i=1}^{n} p_\theta(x_i|x_{<i}, c)$

Dataset: $D = \{(x^1, c^1), \dots, (x^{|D|}, c^{|D|})\}$

Decoding: $x^* = \underset{x}{\text{argmax}} \sum_i \log p_\theta(x_i|x_{<i}, c)$

01 文本生成任务简介

Overview & Formulation

2. Non-Autoregressive Generation

$$p_\theta(x|c) = \begin{cases} \prod_{i=1}^{n} p_\theta(x_i|x_{<i}, c) & \text{Autoregressive} \\ \prod_{i=1}^{n} p_\theta(x_i|c) & \text{Non-Autoregressive} \end{cases}$$

$$p_\theta(x|c) = \int p_\theta(x|z,c)p_\theta(z|c)dz$$

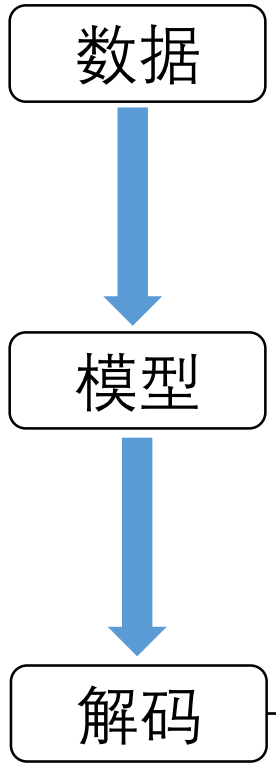
$$p_\theta(x|z,c) = \prod_{i=1}^{n} p_\theta(x_i|z,c)$$

- Inference Speed
- Local optimal search results
- Lack of Context (e.g., Grammar Error Correction)
- ……

OUTLINE

02 文本生成解码算法

- Data Augmentation
- Data Selection



Seq2Seq/VAE/GAN

……

$p_\theta(x|c)$

$$x^* = \arg\max_x p_\theta(x|c)$$

搜索空间大小: $|V|^T$

Sampling
- Top-k Sampling
- Top-p Sampling ……

Beam Search
- Diverse Beam Search
- Group Beam Search
- Stochastic Beam Search

……

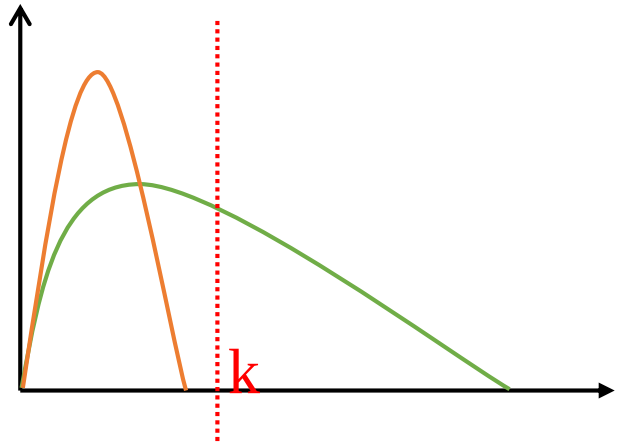
02 文本生成解码算法

Sampling

$$x = x_1, x_2, \ldots, x_t, \ldots, x_n \qquad x^* = \arg\max_x p_\theta(x|c)$$
$$\approx \{\arg\max_{x_t} p_\theta(x_t|x_{<t}, c)\}_{t=1}^{n}$$

1. Top-k Sampling

每一个step t, 从概率$p_\theta(x_t|x_{<t}, c)$最大的k个候选token中随机选取1个



02 文本生成解码算法

2. Top-p Sampling / Nucleus Sampling (Holtzman et al., 2020)

每一个step t, 从概率$p_\theta(x_t|x_{<t},c)$最大的M候选token中随机选取1个

其中M 满足:

$$\sum_{m=1}^{M} p_\theta(x_t = w_m|x_{<t},c) \geq p$$

$$x^* = \arg\max_x p_\theta(x|c)$$

- 贪心策略，解空间中沿一条随机路径的搜索
- 采样出的句子质量难以保证
  通顺性、BLEU、ROUGE

Holtzman et al. The Curious Case of Neural Text Degeneration. ICLR 2020.

02 文本生成解码算法

Beam Search

1. Naïve Beam Search

- a limited-width breadth first search
- stores the top-B highest scoring partial solutions at each time step

Beam Width/ Beam Size

x $x_{[t-1]} = x_1, x_2, \dots, x_{t-1}$

Beam candidates/hypotheses

$$X_{[t-1]} = \{x_{1,[t-1]}, x_{2,[t-1]}, \dots, x_{B,[t-1]}\}$$

每一步t, 为当前每个Beam candidate扩展一个token

$$\Gamma_t = \{x | x_{[t-1]} \in X_{[t-1]} || w \in V\}$$

$$|\Gamma_t| = B * |V|$$

$$X_{[t]} = \arg topB\, G(x_{[t]}), x_{[t]} \in \Gamma_t$$

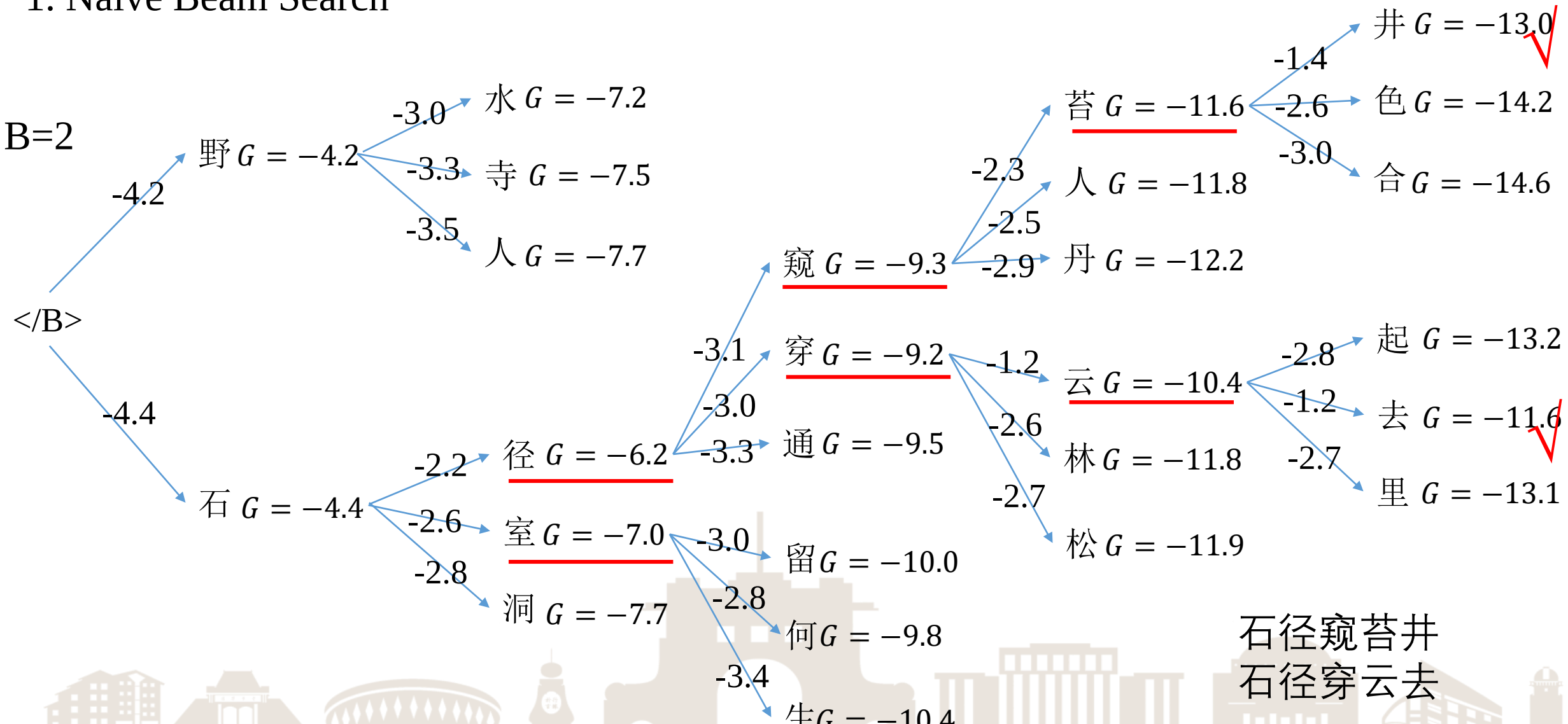
$$x_{j,[t]} \neq x_{i,[t]}, \forall i \neq j$$

Beam Score

$$G(x_{[t]}) = \sum_{i=1}^{t} \log p_\theta(x_i | x_{[i-1]}, c)$$

02 文本生成解码算法

1. Naïve Beam Search



石径窥苔井
石径穿云去

02 文本生成解码算法

1. Naïve Beam Search

t=1

```
0, 水, 4.602
1, 草, 4.935
2, 石, 4.403
3, 远, 4.790
4, 不, 4.947
5, 白, 4.696
6, 秋, 4.942
7, 独, 4.843
8, 山, 4.479
9, 孤, 4.908
10, 江, 4.739
11, 客, 4.743
12, 春, 4.836
13, 日, 4.680
14, 古, 4.782
15, 云, 4.691
16, 野, 4.179
17, 去, 5.013
18, 落, 5.079
19, 林, 5.095
```

t=2

```
0, 日暮, 6.194
1, 草色, 5.861
2, 石径, 6.693
3, 落日, 5.556
4, 远树, 6.526
5, 草径, 7.010
6, 古寺, 6.663
7, 野水, 7.152
8, 落叶, 6.996
9, 客路, 5.942
10, 去去, 6.940
11, 孤城, 6.622
12, 白日, 7.075
13, 古木, 6.954
14, 石室, 7.037
15, 日落, 6.643
16, 白云, 6.798
17, 独坐, 7.205
18, 去国, 7.262
19, 孤云, 7.213
```

t=3

```
0, 古寺无, 9.034
1, 远树带, 8.728
2, 落日临, 9.054
3, 日落沙, 9.038
4, 远树连, 8.974
5, 落日辞, 8.056
6, 客路逢, 8.928
7, 客路依, 8.284
8, 落日孤, 8.031
9, 客路多, 9.133
10, 草色连, 8.035
11, 草色青, 9.030
12, 客路青, 9.138
13, 古木寒, 9.379
14, 落叶辞, 9.303
15, 落日枫, 9.364
16, 落日归, 9.373
17, 落日荒, 9.143
18, 客路随, 9.214
19, 日暮长, 9.414
```

t=4

```
0, 日暮长沙, 10.230
1, 落日辞林, 10.412
2, 落日孤峰, 10.497
3, 落日临水, 10.475
4, 落日枫林, 9.777
5, 古寺无人, 10.500
6, 落日孤飞, 9.931
7, 落日孤山, 10.281
8, 远树连天, 10.439
9, 草色连湖, 10.517
10, 落日孤云, 10.526
11, 远树带烟, 10.768
12, 落日辞荒, 10.676
13, 草色连天, 10.550
14, 客路青山, 10.728
15, 草色连云, 10.734
16, 落日孤戍, 10.685
17, 古寺无多, 10.854
18, 客路依林, 10.862
19, 落日辞姜, 10.863
```

t=5

```
0, 草色连湖水, 11.280
1, 客路青山外, 12.177
2, 落日孤飞鹜, 10.072
3, 古寺无多路, 11.969
4, 客路依林下, 12.567
5, 落日临水岸, 12.380
6, 客路青山在, 12.219
7, 落日枫林上, 12.525
8, 远树连天未, 12.496
9, 落日枫林外, 11.521
10, 落日孤山外, 12.233
11, 落日辞林下, 12.162
12, 远树连天碧, 12.202
13, 落日辞林去, 12.288
14, 草色连天暝, 12.572
15, 落日辞林苑, 12.056
16, 古寺无人到, 12.588
17, 落日孤戍外, 11.044
18, 落日孤峰合, 11.831
19, 古寺无人迹, 12.479
```

02 文本生成解码算法

2. Diverse Beam Search (Li et al., 2016)

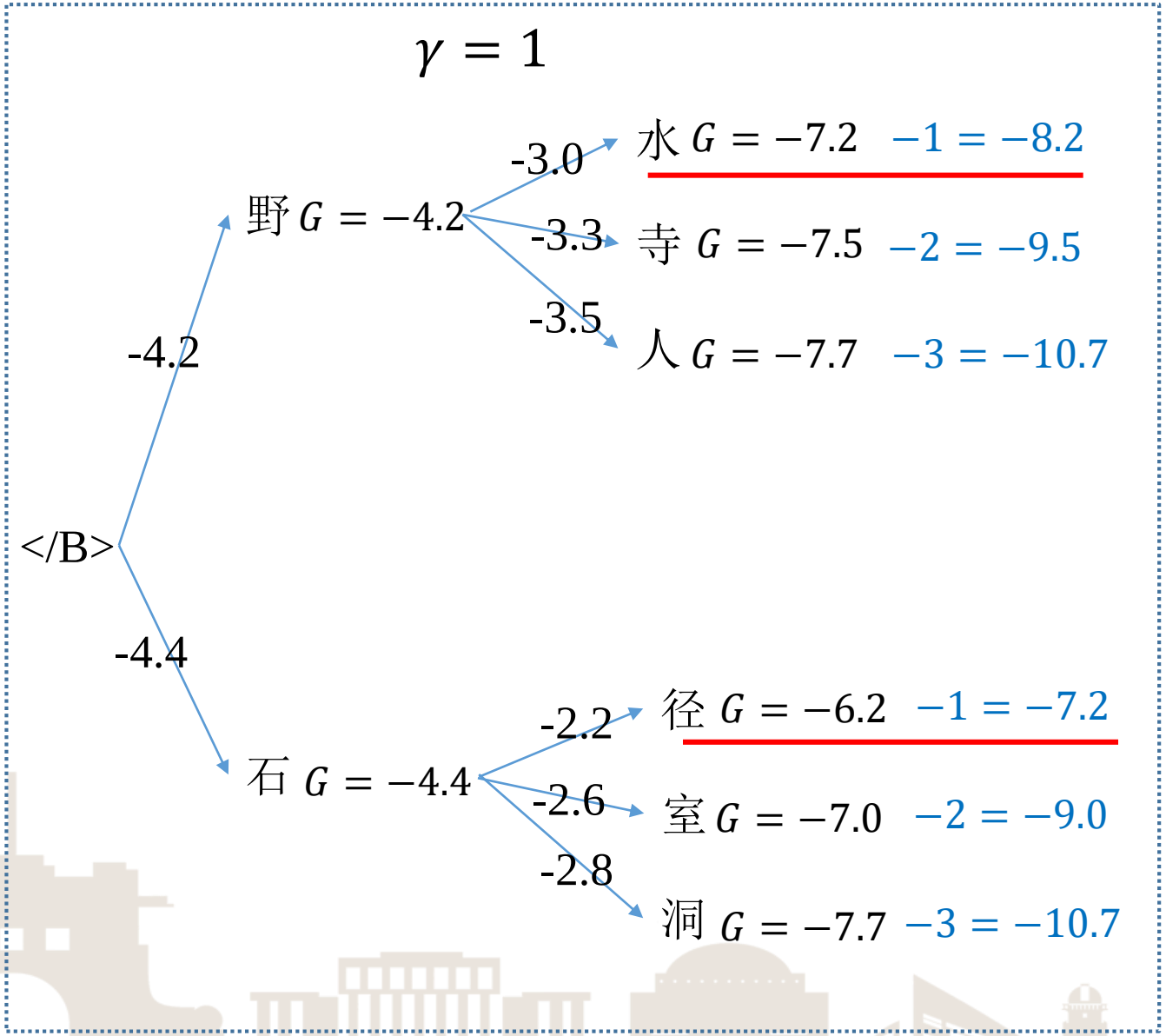
$$G(x_{[t]}) = \sum_{i=1}^{t} \log p_\theta(x_i | x_{[i-1]}, c)$$

$$= G(x_{[t-1]}) + \log p_\theta(x_t | x_{[t-1]}, c)$$

$$= G(x_{[t-1]}) + \log p_\theta(x_t | x_{[t-1]}, c) - \gamma * k$$

γ: diversity rate

k: the ranking of the current hypothesis among its siblings



Li et al. A Simple, Fast Diverse Decoding Algorithm for Neural Generation. 2016.

02 文本生成解码算法

2. Diverse Beam Search (Li et al., 2016)

γ: diversity rate

$$G(x_{[t-1]}) + \log p_\theta(x_t | x_{[t-1]}, c) - \gamma * k$$

Automatically Learning Diversity Rate

$$\pi(\gamma(c) = \gamma' | c) = \frac{\exp(h_c^T \cdot h_{\gamma'})}{\sum_\gamma \exp(h_c^T \cdot h_\gamma)}$$

γ与前序子串$x_{[t-1]}$有关?
γ与当前位置t有关?

Li et al. A Simple, Fast Diverse Decoding Algorithm for Neural Generation. 2016.

02 文本生成解码算法

3. Group Beam Search (Vijayakumar et al., 2018)

Diversity: 在任意t，要求当前hypotheses集合$X_{[t]} = \{x_{1,[t]}, x_{2,[t]}, \dots, x_{B,[t]}\}$中的序列尽可能两两不相同

将hypotheses分为M组，每组$B' = \frac{B}{M}$个，t时刻第g组，$g = 1,2,..,M$ 为$X^g_{[t]} = \{x^g_{1,[t]}, \dots, x^g_{B',[t]}\}$

$$G(x_{[t]}) = \sum_{i=1}^{t} \log p_\theta(x_i | x_{[i-1]}, c)$$

$$G'\left(x^g_{[t]}\right) = G\left(x^g_{[t]}\right) + \lambda \sum_{h=1}^{g-1} \Delta(x^g_{[t]}, X^h_{[t]})$$

diversity function

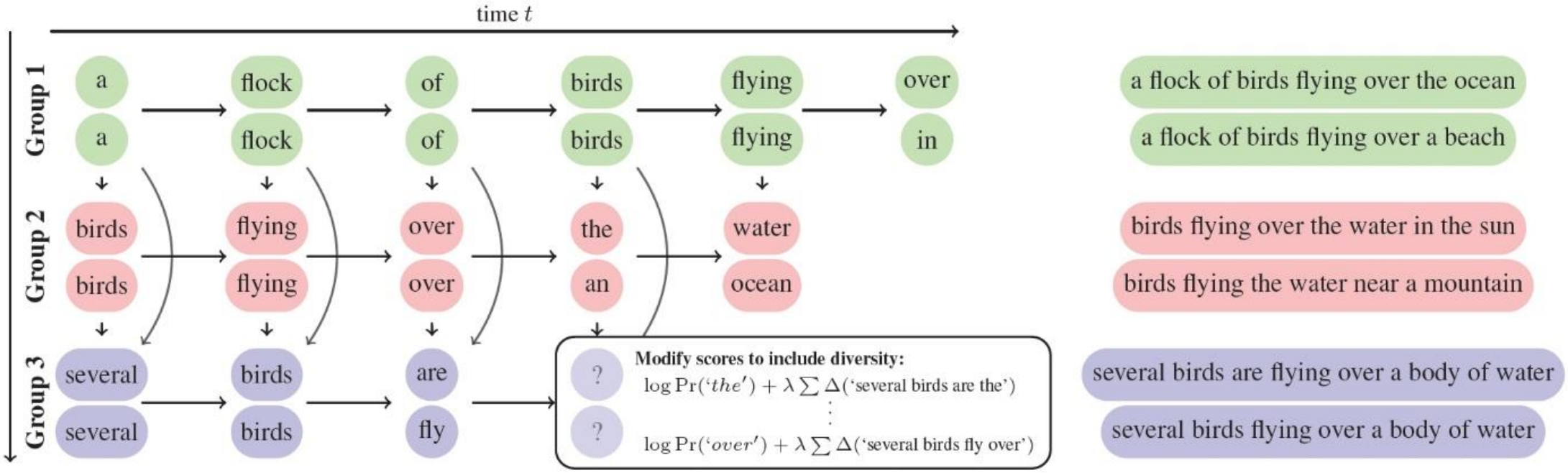
$$\Delta\left(x^g_{[t]}, X^h_{[t]}\right) = \sum_{b=1}^{B'} \delta(x^g_{[t]}, x^h_{[t]})$$

Dissimilarity measure

Vijayakumar et al. Diverse Beam Search for Improved Description of Complex Scenes. AAAI 2018.

02 文本生成解码算法

3. Group Beam Search (Vijayakumar et al., 2018)



组内hypotheses并行扩展，组间有先后依赖关系！

- $M = B$, diversity最好；各hypotheses完全顺序依赖，速度最慢
- $M = 1$, 退化为原Beam search；diversity无提升，速度最快

Vijayakumar et al. Diverse Beam Search for Improved Description of Complex Scenes. AAAI 2018.

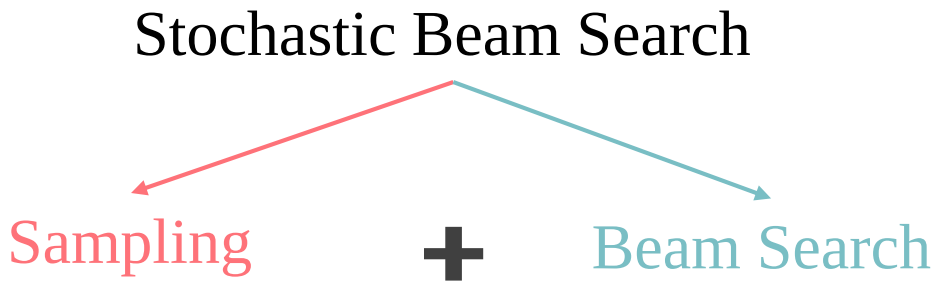
02 文本生成解码算法

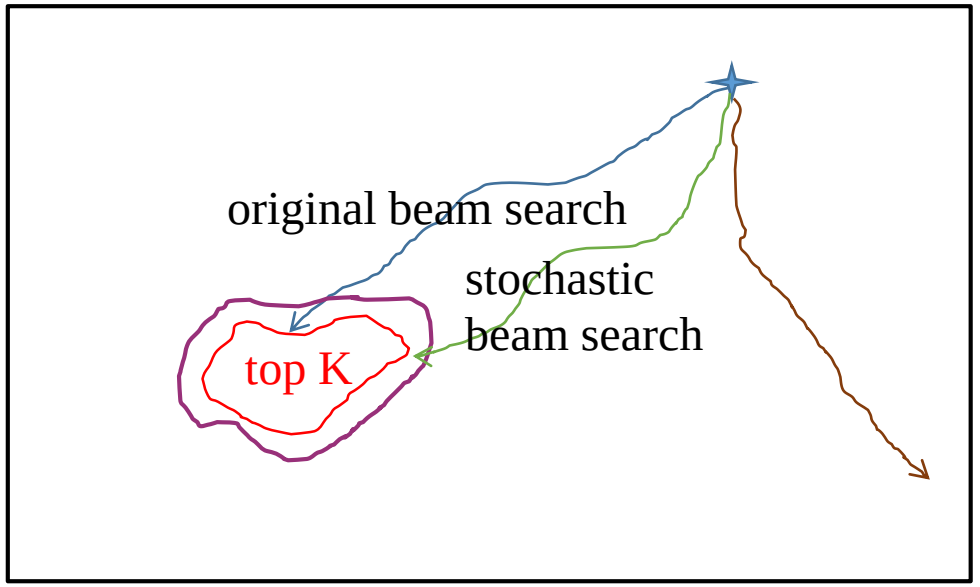
4. Stochastic Beam Search (Kool et al., 2019)

$$X_{[t]} = \arg topK\ G(x_{[t]}), x_{[t]} \in \Gamma_t$$

- 增加diversity
- 速度与原始Beam Search相比没有明显降低

直接将top K 换为从top K中sample？





Kool et al. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. ICML 2019.

02 文本生成解码算法

4. Stochastic Beam Search (Kool et al., 2019)

y_i a sequence $y_{i,j}$ j-th token in y_i

$$G(y_i) = \sum_{j=1}^{t} \log p_\theta(y_{i,j}|y_{i,<j}, c)$$

$$G(y_i) = G_{\emptyset_i} \sim Gumbel(\emptyset_i)$$

$$Gumbel(\emptyset_i) = \emptyset_i - \log(-\log U)$$

$$U \sim Uniform(0,1)$$

$$\emptyset_i = \log p_\theta(y_i|c)$$

Theorem 1. *For $k \leq n$, let $I_1^*, ..., I_k^* = \arg\operatorname{top} k\, G_{\phi_i}$. Then $I_1^*, ..., I_k^*$ is an (ordered) sample without replacement from the Categorical $\left(\frac{\exp \phi_i}{\sum_{j \in N} \exp \phi_j}, i \in N\right)$ distribution, e.g. for a realization $i_1^*, ..., i_k^*$ it holds that*

$$P\left(I_1^* = i_1^*, ..., I_k^* = i_k^*\right) = \prod_{j=1}^{k} \frac{\exp \phi_{i_j^*}}{\sum_{\ell \in N_j^*} \exp \phi_\ell} \qquad (4)$$

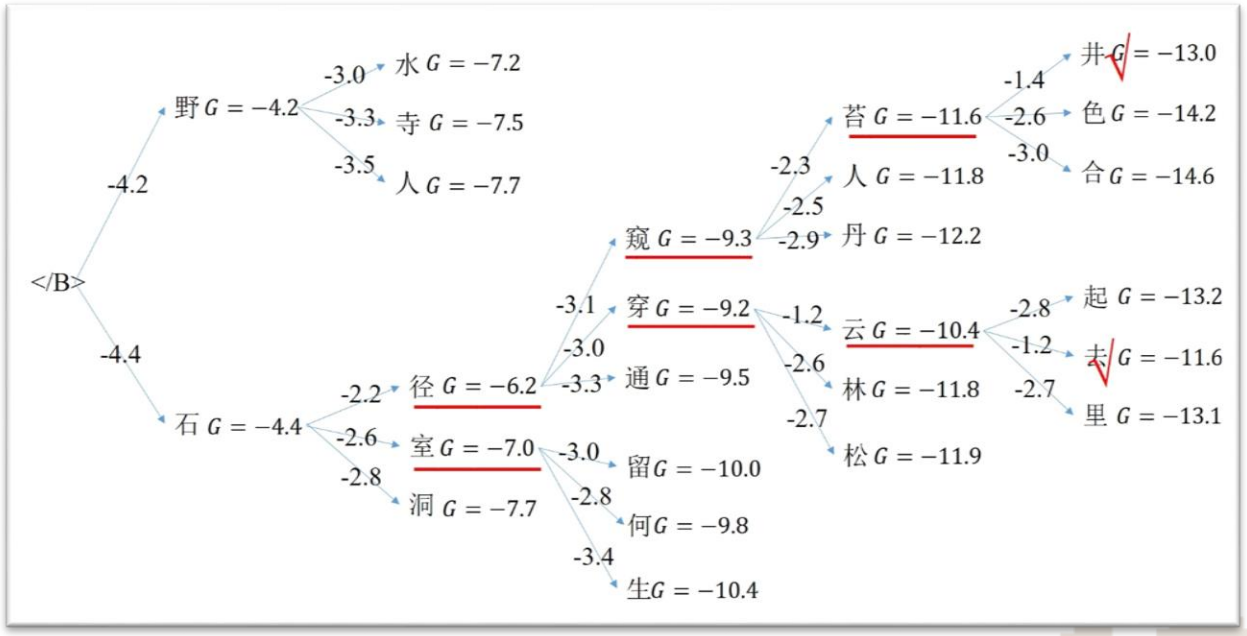
where $N_j^* = N \setminus \{i_1^*, ..., i_{j-1}^*\}$ is the domain (without replacement) for the j-th sampled element.

Kool et al. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. ICML 2019.

02 文本生成解码算法

4. Stochastic Beam Search (Kool et al., 2019)

y_i $G_{\emptyset_i}$



每个叶子节点i对应解空间中的一个句子 y_i

每个中间节点s对应一个前序子串y^s

目标：从根节点向下搜索，确保到达$G_{\emptyset_i}$值最大的k个叶子节点！

$$G_{\phi_S} = \max_{i \in S} G_{\phi_i} \sim \text{Gumbel}(\phi_S)$$

$$G_{\phi_S} = \max_{S' \in \text{Children}(S)} G_{\phi_{S'}}$$

Kool et al. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. ICML 2019.

02 文本生成解码算法

4. Stochastic Beam Search (Kool et al., 2019)

$$G_{\phi_S} = \max_{i \in S} G_{\phi_i} \sim \text{Gumbel}(\phi_S)$$

$$G_{\phi_S} = \max_{S' \in \text{Children}(S)} G_{\phi_{S'}}$$

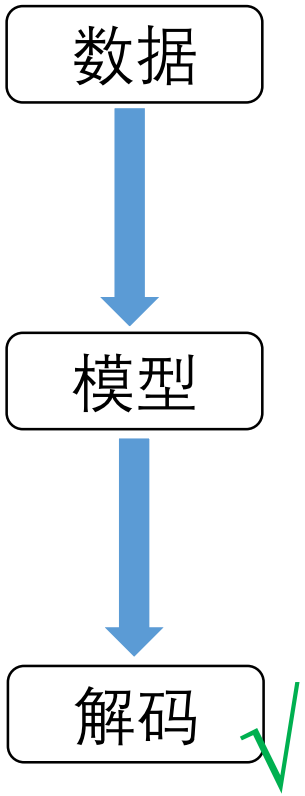
在树的每一层t, 选择$G_{\emptyset_s}$值最大的子节点扩展即可！

Algorithm 1 StochasticBeamSearch($p_{\boldsymbol{\theta}}$, k)

Input: one-step probability distribution $p_{\boldsymbol{\theta}}$, beam/sample size k
Initialize BEAM empty
add $(\boldsymbol{y}^N = \varnothing, \phi_N = 0, G_{\phi_N} = 0)$ to BEAM
for $t = 1, \ldots,$ steps **do**
  Initialize EXPANSIONS empty
  for $(\boldsymbol{y}^S, \phi_S, G_{\phi_S}) \in$ BEAM **do**
    $Z \leftarrow -\infty$
    for $S' \in$ Children(S) **do**
      $\phi_{S'} \leftarrow \phi_S + \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^{S'} | \boldsymbol{y}^S)$
      $G_{\phi_{S'}} \sim \text{Gumbel}(\phi_{S'})$
      $Z \leftarrow \max(Z, G_{\phi_{S'}})$
    end for
    for $S' \in$ Children(S) **do**
      $\tilde{G}_{\phi_{S'}} \leftarrow -\log(\exp(-G_{\phi_S}) - \exp(-Z) + \exp(-G_{\phi_{S'}}))$
      add $(\boldsymbol{y}^{S'}, \phi_{S'}, \tilde{G}_{\phi_{S'}})$ to EXPANSIONS
    end for
  end for
  BEAM $\leftarrow$ take top k of EXPANSIONS according to $\tilde{G}$
end for
Return BEAM

Kool et al. Stochastic Beams and Where to Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. ICML 2019.

02 文本生成解码算法

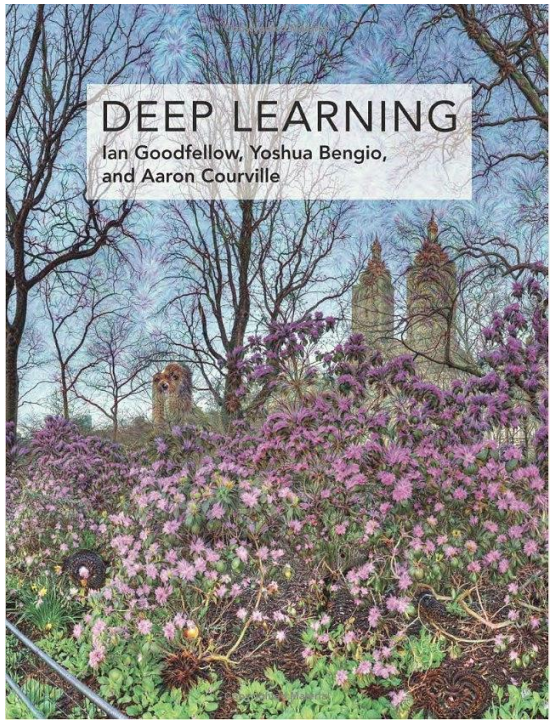


- BERT/GPT等基于大规模语料的pre-training模型训练成本过大
- 预训练模型能够从大语料中学到隐含的语言知识

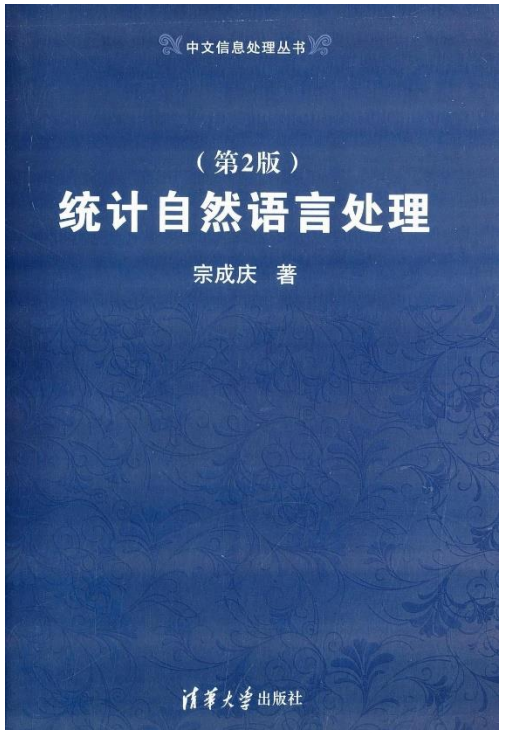
OUTLINE

03 相关资源推荐

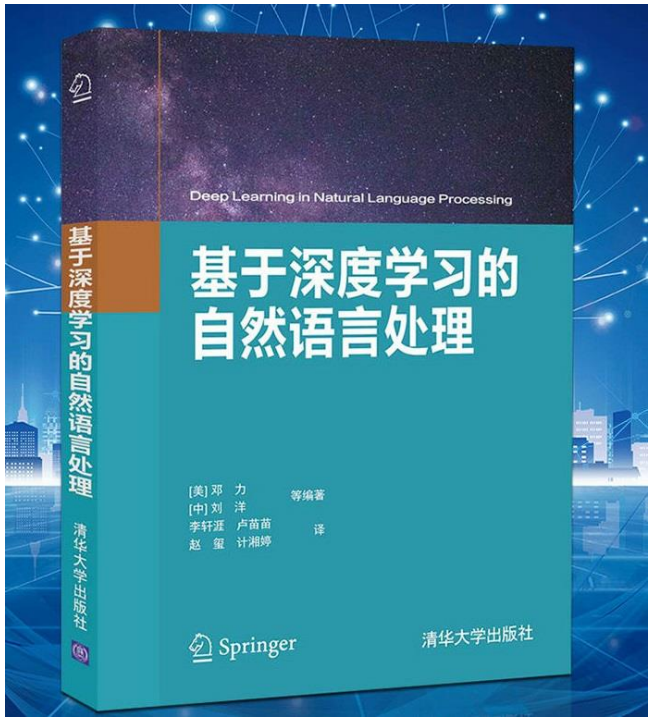
◘学习资料



Deep Learning
Ian Goodfellow and Yoshua Bengio and Aaron Courville



《统计自然语言处理》
宗成庆



《基于深度学习的自然语言处理》
邓力 刘洋

03 相关资源推荐

◘学习资料

Garbacea and Mei, *Neural Language Generation: Formulation, Methods, and Evaluation*

https://arxiv.org/abs/2007.15780

Zhou et al., *Progress in Neural NLP: Modeling, Learning, and Reasoning*

https://www.sciencedirect.com/science/article/pii/S2095809919304928

黄民烈, *Controllable Text Generation: Types, Knowledge, and Logic*

http://coai.cs.tsinghua.edu.cn/hml/media/files/controllable-text-generation.pdf

Lili Mou, Olga Vechtomovaert, *ACL 2020 Stylized Text Generation Tutorial*

https://sites.google.com/view/2020-stylized-text-generation/tutorial

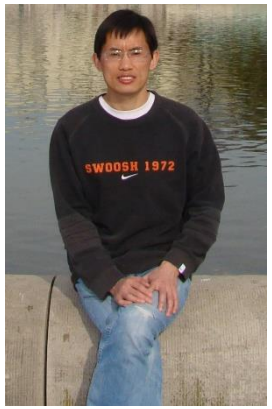
03 相关资源推荐

领域研究者介绍

Xiaojun Wan

Peking University

https://wanxiaojun.github.io/



Minlie Huang

Tsinghua University

http://coai.cs.tsinghua.edu.cn/hml/

Lili Mou

University of Alberta

https://lili-mou.github.io/

03 相关资源推荐

◘THUNLP Lab　清华大学自然语言处理与社会人文计算实验室

自然语言处理与社会人文计算实验室 TSINGHUA UNIVERSITY

http://nlp.csai.tsinghua.edu.cn/

微信公众号： TsinghuaNLP

孙茂松 教授

研究方向为自然语言处理、中文信息处理、Web智能、社会人文计算和计算教育学等

刘洋 教授
研究方向为自然语言处理、机器翻译

刘知远 副教授
研究方向为知识图谱与语义计算、 社会计算与计算社会科学

03 相关资源推荐

Jiuge System (九歌): an Online Chinese poetry generation system

Online system https://jiuge.thunlp.org/

GitHub https://github.com/thunlp-aipoet

- A paper list for the interdisciplinary field of AI and poetry
  https://github.com/THUNLP-AIPoet/PaperList
- Chinese Poetry Datasets
  https://github.com/THUNLP-AIPoet/Datasets

Thanks!

Xiaoyuan Yi　PhD Student

THUNLP Lab, Tsinghua University

Mail: yi-xy@mails.tsinghua.edu.cn
Homepage：https://xiaoyuanyi.github.io/

Any questions or suggestions, please feel free to email me!